



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Detecting People Looking at Each Other in Videos

Citation for published version:

Marin-Jimenez, M, Zisserman, A, Eichner, M & Ferrari, V 2014, 'Detecting People Looking at Each Other in Videos', *International Journal of Computer Vision*, vol. 106, no. 3, pp. 282-296.
<https://doi.org/10.1007/s11263-013-0655-7>

Digital Object Identifier (DOI):

[10.1007/s11263-013-0655-7](https://doi.org/10.1007/s11263-013-0655-7)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

International Journal of Computer Vision

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Detecting people looking at each other in videos

M.J. Marin-Jimenez, A. Zisserman, M. Eichner, V. Ferrari

Received: date / Accepted: date

Abstract The objective of this work is to determine if people are interacting in TV video by detecting whether they are looking at each other or not. We determine both the temporal period of the interaction and also spatially localize the relevant people. We make the following four contributions: (i) head detection with implicit coarse pose information (front, profile, back); (ii) continuous head pose estimation in unconstrained scenarios (TV video) using Gaussian Process regression; (iii) propose and evaluate several methods for assessing whether and when pairs of people are looking at each other in a video shot; and (iv) introduce new ground truth annotation for this task, extending the TV Human Interactions Dataset [28]. The performance of the methods is evaluated on this dataset, which consists of 300 video clips extracted from TV shows. Despite the variety and difficulty of this video material, our best method obtains an average precision of 87.6% in a fully automatic manner.

Keywords person interactions, video search, action recognition, head pose estimation

M.J. Marin-Jimenez
Department of Computing and Numerical Analysis,
Maimonides Institute for Biomedical Research (IMIBIC),
University of Cordoba
Tel.: +34-957212172
Fax: +34-957218360
E-mail: mjmarin(AT)uco.es

A. Zisserman
University of Oxford
E-mail: az(AT)robots.ox.ac.uk

M. Eichner
ETH Zurich
E-mail: marcin.eichner(AT)vision.ee.ethz.ch

V. Ferrari
University of Edinburgh
E-mail: vferrari(AT)staffmail.ed.ac.uk



Fig. 1: **Are they looking at each other?** Answering this question enables richer video analysis, and retrieval based on where actors interact. From left to right: *Friends*, *Casablanca*, *Fawlty Towers*. The eyeline in the *Casablanca* shot gives rise to the famous quote "Here's looking at you, kid".

1 Introduction

If you read any book on film editing or listen to a director's commentary on a DVD, then what emerges again and again is the importance of eyelines. Standard cinematography practice is to first establish which characters are looking at each other using a medium or wide shot, and then edit subsequent close-up shots so that the eyelines match the point of view of the characters. This is the basis of the well known 180° rule in editing.

The objective of this paper is to determine whether eyelines match between characters within a shot – and hence understand which of the characters are interacting. The importance of the eyeline is illustrated by the three examples of fig. 1 – one giving rise to arguably the most famous quote from *Casablanca*, and another being the essence of the humour at that point in an episode of *Fawlty Towers*. Our target application is this type of edited TV video and films. It is very challenging material as there is a wide range of human actors, camera viewpoints and ever present background clutter.

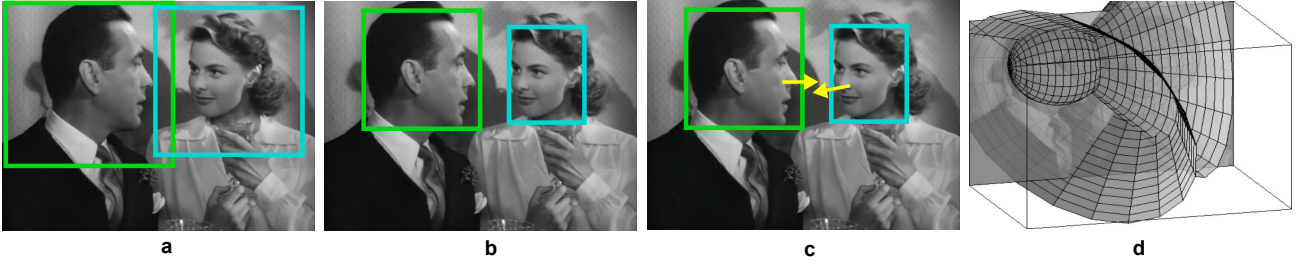


Fig. 2: **Proposed pipeline.** a) Upper-body detection. b) Head detection inside upper-body areas. c) Head pose estimation. d) LAEO scoring between pairs of heads. This example would be correctly classified as LAEO.

Determining whether characters are interacting using their eyelines is another step towards a fuller video understanding, and complements recent work on automatic character identification [7, 11, 37], human pose estimation [1, 6, 17, 33, 46], human action recognition [21, 22, 24, 30, 32], and social [13] and specific interaction recognition (e.g. hugging, shaking hands) [29, 46]. Putting interactions together with previous character identification work, it now becomes possible to retrieve shots where two particular actors interact, rather than just shots where the actors are present in the same scene.

In order to determine if two people are looking at each other, it is necessary to detect their head and estimate their head pose. There are two main strands in previous work: *2D* approaches, where detectors are built for several aspects of the head (such as frontal and profile [37]) or the pose is classified into discrete viewpoints [4, 39, 47], or regressed [26]. The alternative are *3D* approaches, where a 3D model is fitted to the image and hence the pose determined [5, 10]. A survey of head pose estimation is given in [25].

In this work, we start by detecting human heads in each video frame separately and then grouping them over time into tracks, each corresponding to a different person. Next, we estimate the *pitch* and *yaw* angles for each head detection. For this, we propose a 2D approach and train a Gaussian Process regressor [31] to estimate the head pitch and yaw directly from the image patch within a detection window using publicly available datasets. In the third step, we explore three methods to determine if two people (tracks) are Looking At Each Other (*LAEO*, sec. 2). Two people are LAEO if there is eye contact between them. We start with a simple 2D analysis, based on the intersection of gaze areas in 2D defined by the sign of the estimated yaw angles (sec. 2.1). In a more sophisticated alternative, we use both the continuous yaw and pitch angles as well as the relative position of the heads (sec. 2.2). Finally, we propose a ‘2.5D’ analysis, where we use the scale of the detected head to estimate the depth positioning of the actors, and combine it with the full head pose estimate to derive their gaze volumes in 3D (sec. 2.3). Figure 2 summarizes the proposed pipeline.

We apply these methods (sec. 5) to the TV Human Interactions Dataset (TVHID) [28]. This is very challenging video material with far greater variety in actors, shot editing, viewpoint, locations, lighting and clutter than the typical surveillance videos used previously for classifying interactions [3, 27, 40] where there is a fixed camera and scene. We provide additional ground truth annotation for the dataset, specifying which shots contain people looking at each other. Originally, the dataset only had annotations for four specific interactions (hand-shake, high-five, hugging and kissing) but there are many other shots where people are looking at each other.

In a thorough experimental evaluation on the TVHID, we show that the full head pose estimate (i.e. yaw and pitch angles) in combination with the relative position of the heads in a 3D scenario are needed for most real situations to clearly define if two people are LAEO.

This paper is an extended version of our preliminary work on this subject [23].

2 Classifying pairs of heads as looking at each other (LAEO)

Let us assume that we know the spatial location of the persons present in a video sequence and we have information about their head orientation as well. For each person i in a video frame, let

$$W_i = (x_i, y_i, w_i, h_i, \theta_i, \alpha_i, \sigma_{\theta_i}, \sigma_{\alpha_i})$$

be an image window containing the head, with top-left coordinates (x_i, y_i) , width w_i , height h_i , yaw angle θ_i (rotation about Y -axis), pitch angle α_i (rotation about X -axis). The values σ_{θ_i} and σ_{α_i} represent the uncertainty in the estimate of θ_i and α_i , respectively. Using this information, we propose in this section three methods for classifying a pair of persons as LAEO or not. This is the main contribution of this paper. In section 3 we explain how to perform head pose estimation automatically (i.e. we use Gaussian Process regressors), and in section 4 we explain how to detect and tracks



Fig. 3: **Left: Intersection of gaze areas in 2D.** We show heads (P_l and P_r) as red rectangles and gaze areas (G_l and G_r) as yellow dashed rectangles. The head orientations are represented by green arrows (O_l and O_r). This method would incorrectly say that these people are not LAEO, since their 2D gaze areas do not intersect. **Right: Geometric constraints in 2D.** We show the estimated yaw and pitch angles as yellow vectors (yaw θ determines if left or right facing and length; pitch α determines orientation). The blue vector γ_{lr} defines the orientation of the vector going from B to C in the image plane. The angle defined by these vectors for (B,C) would classify such pair as LAEO. (Best viewed in colour).

heads (i.e. we build tracks of head detections obtained from previously computed upper-body tracks).

2.1 Intersection of gaze areas in 2D

The simplest method we propose only considers the head pose as discretized into just two directions, i.e. facing left or right. For this we only use the estimated yaw angle and discard the pitch. In addition to this binary head pose, this method also uses the image position of the head window and its height.

We define as *gaze area* G_i the image region a person head P_i is looking at: a horizontal rectangle extending from the head towards the gaze direction (fig. 3(left)). The height of G_i is given by the height of P_i , while the width is given by the x position of the farthest other head in the scene. To classify whether two heads P_l, P_r are LAEO, we define the $\text{LAEO}_{GA}(P_l, P_r)$ function. Let (x_l, y_l) and (x_r, y_r) be the centres of P_l, P_r , satisfying the condition $(x_l \leq x_r)$, and O_l, O_r be their orientation (i.e. +1 facing left, -1 facing right). With these definitions, LAEO_{GA} is

$$\text{LAEO}_{GA}(P_l, P_r) = \text{IoU}(G_l, G_r) \cdot \delta(O_l \cdot O_r < 0) \quad (1)$$

where $\text{IoU}(G_i, G_j) = \frac{G_i \cap G_j}{G_i \cup G_j}$ is the intersection-over-union of the heads' gaze areas G_i, G_j (fig. 3(left)); the Kronecker delta $\delta(c)$ is 1 if condition c is true, and 0 otherwise.

2.2 Geometric constraints in 2D

The second method we propose takes into account both the yaw and pitch angles defining the full head pose, as well as

the image position of the two heads. Two people are deemed to be LAEO if all the following three conditions are true

- (i) the person on the left has a positive yaw angle and the person on the right has a negative yaw angle
- (ii) the cosine of the difference between their yaw angles is close to -1
- (iii) the vectors defined by the pitch angles are similar to the vectors that join the heads, in both directions.

Fig. 3(right) shows an example that should be highly scored as LAEO.

For a head P_i , let (x_i, y_i) be the coordinates of its centre, θ_i, α_i the estimated yaw and pitch angles, and $\sigma_{\theta_i}, \sigma_{\alpha_i}$ the uncertainty associated at each estimated angle, respectively. We define the following function $\text{LAEO}_{GC}(P_l, P_r)$ to formalize the above constraints and decide if two heads P_l, P_r are LAEO (with $(x_l \leq x_r)$):

$$\begin{aligned} \text{LAEO}_{GC}(P_l, P_r) = & \beta_{\theta} \cdot [\delta(\theta_l \cdot \theta_r < 0 \wedge \theta_l > \theta_r) \cdot (1 - \cos(\theta_l - \theta_r)) \cdot 0.5] + \\ & \beta_{\alpha} \cdot [(1 + \cos(\alpha_l - \gamma_{lr})) \cdot 0.25 + (1 + \cos(\alpha_r - \gamma_{rl})) \cdot 0.25] \end{aligned} \quad (2)$$

where γ_{ij} is the orientation of the vector going from P_i to P_j in the image plane; the symbol ‘-’ between two angles denotes their orientation difference; β_{θ} and β_{α} are weights, so that $\beta_{\theta} + \beta_{\alpha} = 1$. Note that each row of eq. (2) (omitting their β) ranges in $[0, 1]$. Therefore, LAEO_{GC} ranges in $[0, 1]$, with 1 the best possible score.

There are many possible choices for this scoring function, as long as they encode the three conditions stated above. In our case, the first term of eq. (2) encodes conditions (i) and (ii), based on the yaw angles. The second term

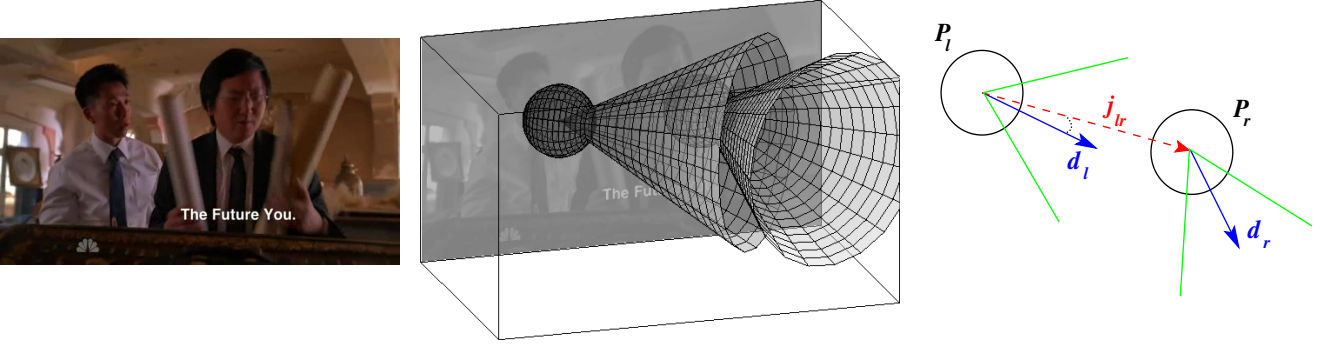


Fig. 4: **Geometric constraints in 3D.** (left) Original video frame. (middle) 3D representation of a scene with two people. We show heads (spheres) and their gaze volumes (cones). (right) View from above, with heads (circles) and gaze direction vectors (blue arrows) \mathbf{d}_l and \mathbf{d}_r defined by the yaw and pitch angles. Green lines are the boundaries of the conic gaze volumes. The red vector is \mathbf{j}_{lr} and goes from P_l to P_r . With this configuration, P_r lays inside P_l gaze area but P_l does not lay inside that of P_r . Therefore, the two people are correctly classified as not LAEO. (Best viewed in colour).

encodes condition (iii), based on the pitch angles and the position of the heads.

The weights β_θ and β_α can be defined as functions of the uncertainties σ_θ and σ_α associated to the angles θ and α , respectively. These uncertainties are output by the Gaussian Process regressors along with the angle estimates themselves (Sec. 3). Hence, we set β_θ for a test pair of heads P_l, P_r as

$$\beta_\theta = (\sigma_{\theta_l}^{-1} + \sigma_{\theta_r}^{-1}) / (\sigma_{\theta_l}^{-1} + \sigma_{\theta_r}^{-1} + \sigma_{\alpha_l}^{-1} + \sigma_{\alpha_r}^{-1}) \quad (3)$$

and $\beta_\alpha = 1 - \beta_\theta$. This dynamic weighting gives more weight to reliable estimates of the head orientation, while reducing the negative impact of poor estimates on the LAEO score.

2.3 Geometric constraints in 3D

The most complex method we propose operates in a simplified 3D space. We place each person’s head P_i in a common 3D coordinate system by using the image coordinates of the head centre as (x_i, y_i) and deriving the depth coordinate z_i from the head size in the image. Coordinates z_i are derived as a direct proportion between all the heads present in the scene, by assuming that in such 3D world all the heads have the same size and, therefore, the height of the detection window indicates the relative distance of the person to the camera (i.e. larger heads in 2D are closer to the camera than smaller ones). So, heads are z -ordered so that the largest head in the image is the closest one to the camera. This implicitly assumes that all people have approximately the same head size in the 3D world. This is only a problem in rare cases, i.e. scenes containing both adults and small children, which have significantly different head sizes.

The gaze volume of a head P_i is represented as a 3D cone C_i with apex at (x_i, y_i, z_i) and axis orientation defined

by the estimated yaw and pitch angles (fig. 4). We classify two heads P_l and P_r as LAEO if P_l lays inside C_r , and P_r lays inside C_l . Note how this method uses all the available information.

More formally, we define the LAEO_{3D} score by the following equation:

$$\text{LAEO}_{3D}(P_l, P_r) = \frac{(\varphi - \Delta(\mathbf{j}_{lr}, \mathbf{d}_l)) + (\varphi - \Delta(\mathbf{j}_{rl}, \mathbf{d}_r))}{2\varphi} \quad (4)$$

where the angle φ represents the *aperture* of the gaze cone that is a free parameter to be learnt during training (see Sec. 5.3); $\Delta(\cdot, \cdot)$ is the angle between two vectors; \mathbf{d}_i is a vector defined by the yaw and pitch angles of P_i ; \mathbf{j}_{lr} is the vector from P_l to P_r , i.e. defined as $(x_l, y_l, z_l) \rightarrow (x_r, y_r, z_r)$ (and vice-versa for \mathbf{j}_{rl}). Fig. 4 illustrates this score. Note how the magnitude of \mathbf{d}_i is irrelevant, as it is only used inside the Δ function.

3 Continuous head pose estimation

We describe here our approach to automatically estimate two head pose angles: *yaw* (around the Y axis) and *pitch* (X axis). We do not consider *roll* (Z axis). We use a Gaussian Process (GP) to directly regress from the image patch within a head detection window to the two pose angles.

3.1 Training a Gaussian Process head pose regressor

For each detected head, we crop an $N \times N$ image window H centred on it, where N is the number of pixels of the largest side of the detection window. Then, H is resized to a predefined common size $N' \times N'$. Given an observed head window H , the goal is to predict two angles (θ, α) conveying its pose with regard to the camera viewpoint. We formulate this problem in terms of regression, and train two separate



Fig. 5: **Head pose datasets.** (top) Samples from CMU-PIE dataset. (bottom) Samples from IDIAP-HP dataset.

regressors, one for yaw (θ) and one for pitch (α). As the method is exactly the same, we restrict the explanation to yaw.

The goal is to find a real-valued regression function $\hat{\theta} = f(g(H))$, so that $\hat{\theta} \approx \theta$, where $g(H)$ is a feature vector of H , and θ and $\hat{\theta}$ are the real and estimated angles respectively. We use a histogram of oriented gradients (HOG) [8] as the head descriptor g . A HOG descriptor encodes the spatial structure of a rather rigid object through a set of histograms of oriented gradients computed in each cell of a grid overlaid on the window covering the object.

A Gaussian Process (GP) [31] regressor $f(g(H))$ is employed for estimating the angle. GPs are attractive because they are non-parametric models, and therefore can flexibly adapt to almost any distribution of the data (i.e. provided the mean, covariance and likelihood functions). Moreover, at inference time, they return both the estimate $\hat{\theta}$ as well as its uncertainty $\sigma_{\hat{\theta}}$ (i.e. the mean and variance of the Gaussian posterior). This offers the possibility to downweight uncertain pose estimates in later processing stages (e.g. sec. 2.2).

3.2 Implementation details and experimental validation

A GP [31] is a collection of random variables, any finite number of which have a joint Gaussian distribution. A GP is completely specified by its mean function m and covariance (or kernel) function k . Given an input vector \mathbf{x} , the mean function $m(\mathbf{x})$ and the covariance $k(\mathbf{x}, \mathbf{x}')$ of a real process $f(\mathbf{x})$ are defined as

$$m(\mathbf{x}) = E[f(\mathbf{x})], \quad (5)$$

$$k(\mathbf{x}, \mathbf{x}') = E[(f(\mathbf{x}) - m(\mathbf{x}))(f(\mathbf{x}') - m(\mathbf{x}'))] \quad (6)$$

where E denotes expectation, and $k(\mathbf{x}, \mathbf{x}')$ indicates that the covariance function is evaluated at the points \mathbf{x} and \mathbf{x}' .

Therefore, we write the GP as

$$f(\mathbf{x}) \sim GP(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')) \quad (7)$$

where $f(\mathbf{x})$ is a stochastic function that is distributed as a GP with parameters m and k . We use a linear mean function $m(\mathbf{x}) = \mathbf{a}\mathbf{x}^T + c$, where \mathbf{a} and c are the *hyperparameters* of m . We investigate experimentally various functional forms for the covariance k below.

For making predictions (i.e. computation of the posterior), one also needs to define the functional form of the likelihood function. A common choice is a Gaussian likelihood, since it allows to carry out exact inference [31].

We use two datasets to learn yaw and pitch angles. The first is the *CMU Pose, Illumination and Expression (CMU-PIE)* dataset [35]. It contains images of 68 people from 13 different camera viewpoints, corresponding to 9 discretized yaw angles ($[-90, 90]$ degrees). Images have been captured in two different sessions and in each session there are four subsets, corresponding to different types of variations: *expression*, *illumination*, *lighting* and *talking*. The top row of Figure 5 shows some examples of images contained in the dataset.

The second dataset is the *IDIAP head pose (IDIAP-HP)* [2]. It contains 8 videos recorded in a meeting room and 15 videos in an office. Yaw, pitch and roll angles ground-truth is provided for each person in every frame. The bottom row of Figure 5 shows some examples of cropped frames from videos of the *meeting room* subset. Note that, in contrast to CMU-PIE dataset, people pose diverse pitch angles.

In order to train the head pose estimators, the first step is to detect all the heads from the training images by using the detector of sec. 4.1. Next, all detected heads are normalized to a common size of 48×48 pixels and HOG features are extracted. For computing the HOG descriptor, we use non overlapping cells of 8×8 pixels and 9 orientation bins for quantizing the orientation of the gradient vectors. We experimented with other configurations for HOG, but they brought

	$SEiso$ $\sigma_f^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')\right)$	$SEisoWN$ $\sigma_f^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')\right) + \sigma_{\mu'}^2 \delta(\mathbf{x}-\mathbf{x}')$	$SEisoPoly$ $\sigma_f^2 \exp\left(-\frac{1}{2l^2}(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')\right) + \sigma_{\mu'}^2 (\mathbf{x}^T \mathbf{x}' + c)^3$	$RQiso$ $\sigma_f^2 \left(1 + \frac{1}{2al^2}(\mathbf{x}-\mathbf{x}')^T(\mathbf{x}-\mathbf{x}')\right)^{-\alpha}$	Lin $(\mathbf{x}^T \mathbf{x}' + 1)/l^2$
yaw	15.73	20.42	20.36	20.40	25.86
pitch	10.09	10.10	10.11	10.16	10.94

Table 1: **RMSE for yaw and pitch.** Average RMSE performance for various functional forms. The minimum values (i.e. lowest error) for each angle are marked in bold. See main text for discussion.

no improvement. Moreover, this configuration is the same as the one used by our head detector, enabling to reuse previous computations. The HOG features are used as input \mathbf{x} to the GP regressor, which outputs the target angle (i.e. θ or α). We learn the parameters of the two GP regressors by using the GPML 3.1 library [43].

We learn the yaw estimator from the subsets *expression* and *illumination* of CMU-PIE dataset, and the pitch estimator from the subset *meeting room* of IDIAP-HP dataset. The set of training data for yaw angle is $\mathcal{D} = \{(g(H_i), \theta_i)\}$, where $g(H_i)$ is the HOG descriptor of the i -th training sample (i.e. head) and θ_i is its ground-truth yaw angle. In order to evaluate the yaw GP regressor, we split the dataset in two parts: six random people are used for validation and the remaining ones for training. We have repeated this procedure for five trials. We measure performance as the root mean squared error (RMSE) averaged over all validation sets, where the error is measured as the difference between the ground-truth angle and the estimated one. We repeat the same procedure for the pitch GP regressor, but using only one for validation in each trial, and all others for training.

We tested various functional forms for the covariance function: diagonal squared exponential ($SEiso$), $SEiso$ plus white noise ($SEisoWN$), $SEiso$ plus third-order polynomial ($SEisoPoly$), rational quadratic ($RQiso$), and linear with bias (Lin). The number of hyperparameters that have to be learned for each covariance function is different: 2 for $SEiso$, 3 for $SEisoWN$, 4 for $SEisoPoly$, 3 for $RQiso$ and 1 for Lin . Each entry in Table 1 reports the regression performance for a particular covariance function and angle. Note how $SEiso$ leads to the best performance for both angles (yaw and pitch), i.e. 15.73 average RMSE for yaw and 10.09 for pitch. The worst results are delivered by a linear covariance function (i.e. Lin).

As a baseline, we trained and validated a linear regressor on the same data (using Matlab’s *robustfit* function). This linear regressor has about twice the average RMSE of the GP with $SEiso$ covariance. This demonstrates that GP regression is a much better choice than simple linear regression for this task.

After the above evaluations, we chose GP regression with a $SEiso$ covariance and we trained a final GP regressor from all the available samples. This final regressor is used in the LAEO experiments (sec. 5).

4 Detecting and tracking heads in video shots.

We explain here how we detect and track the heads of the people present in a video shot. This is in fact the first step in our processing pipeline. We split the task into the following subtasks: (i) human upper-body detection in individual frames; (ii) grouping upper-body detections over time into tracks; (iii) detecting heads within upper-body detection areas; and, (iv) grouping head detections into tracks.

We propose this two-level pipeline because upper-body detection is more robust to clutter than head detection, as it benefits from wider, more distinctive context. The precise localization of the head within the limited area defined by an upper-body detection can then proceed safely. In particular, direct detection of profile heads in uncontrolled scenes would otherwise produce many false positives [18]. On the other hand, although we already have tracks in step (ii), another tracking stage is performed in step (iv) in order to resolve situations where two heads are so close that they fall into the same upper-body bounding box (e.g. see Fig. 8, bottom-left).

The detectors are described next, followed by the tracking process in Sec. 4.2.

4.1 Upper-body and head detection

For both the upper-body (UB) and head detectors we use the model of Felzenszwalb et al.[14], and train using the code released at [42]. This code automatically learns the actual components of the detector based on the aspect ratio of the annotated bounding boxes of the positive training samples. However, it is necessary to indicate the number of desired components. In our case, we set one component for the upper-body detector (without a mirror component), two components for the frontal/profile head detector (plus the corresponding mirror ones) and one component for the back-view head detector (plus the mirror one). Figure 7 shows the root filters of the components generated by the learning code.

Figure 8 shows examples of UB and head detections in a variety of situations (i.e. different viewpoints, scales, illumination, clothing, clutter, ...).



Fig. 6: **Examples of samples used for detector training.** Left: for upper-body detector; right: for head detector. Note the variety of poses, people, clothing (in the UB case), and rear of heads.

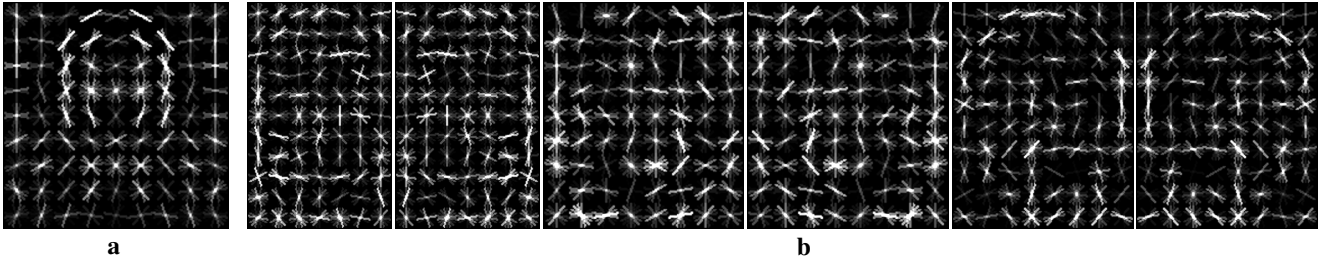


Fig. 7: **Models for the multi-view upper-body and head detectors.** (a) Root filter of the UB detector. This model contains a single component trained from a mixture of all viewpoints. (b) Root filters of the 6 components of the head detector. Each component provides coarse information about the head orientation. From left to right: two near frontal viewpoints, two profile viewpoints and two back viewpoints.

Training and implementation details. We have used a total of 1122 annotated video frames from Hollywood movies [20] as positive training samples for the upper-body detector. These contain upper-bodies viewed from different viewpoints and at different scales. Some examples of UB used during training are shown in Figure 6 (left). As negative training samples, we used those images in the INRIA-person dataset [41] which do not contain people.

The very same set of Hollywood video frames has been used for training the frontal/profile view components of the head detector. Since the Hollywood movies dataset contains very few back-views of heads, the positive training set for the back-view components of the head detector are manually annotated on 199 video frames extracted from the IDIAP head pose dataset [2]. Some examples of heads used for training are shown in Figure 6 (right). As this head detector is intended to be run only inside upper-body windows, we provide negative training samples from the area surrounding the head.

4.2 Person tracking

We describe here the tracker we use to connect over time the single-frame detections produced in the previous stage. The same tracker is used to track upper-body detections or head detections.

For this we design a tracker that combines successful ideas from recent works. As in [11], detections in different frames that are connected by many KLT point tracks [34] are more likely to be grouped in the same track. As in [36] for faces, we exploit the fact that detections with similar appearance are more likely to be the same person and therefore should be grouped even if far away in time. This helps recovering from full occlusion. Finally, we borrow from [16] the idea of casting the tracking process as a clique partitioning problem. This provides a clear objective function and a well-explored approximate minimization algorithm.

Affinity measures. More formally, we combine three different kinds of features as cues for grouping: (i) the location of a detection window, (ii) its appearance, and (iii) the motion of point tracks inside it. We measure the affinity between every pair of detections D_i, D_j in the whole shot according to each of these features.

The location affinity $W_{loc}(i, j)$ is computed as the area of intersection-over-union between D_i and D_j . Note how this takes into account both the position and the scale of the detections. This was the only affinity term used in our previous work [16].

The appearance of each detection is represented by a normalized LAB color histogram. The appearance affinity between two detections D_i, D_j is based on the Euclidean dis-

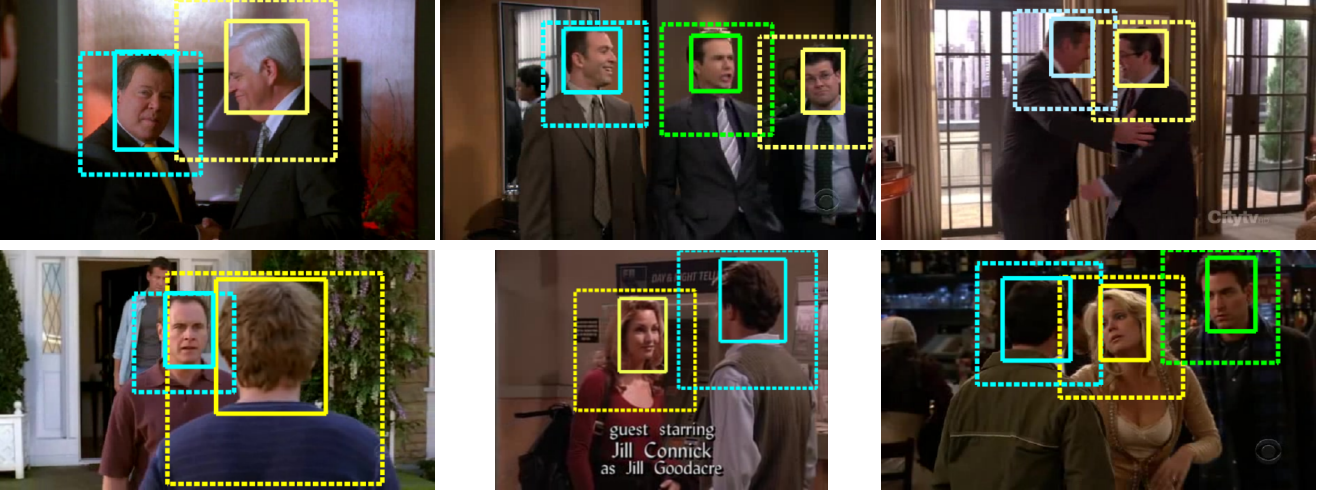


Fig. 8: **Examples of UB (dashed) and head (solid) detections.** The head detector is only run inside UB detection windows. Note how heads are localised in various relative positions within the UB windows, adapting to the image content, including back views.

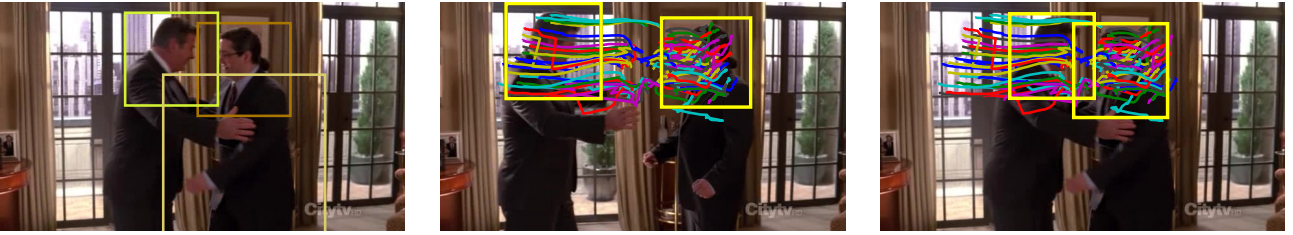


Fig. 9: **Person tracking - motion affinity term.** (left) Input upper-body detections in a video frame. (middle, right) KLT point tracks are one of the three cues we use to robustly group detections into tracks.

tance $E(i, j)$ between their LAB histograms. The final appearance affinity matrix is $W_{app}(i, j) = (2 - E(i, j))/2$.

The last affinity measure counts how many KLT point tracks that pass through D_i also pass through D_j . More precisely, let S_k be the set of KLT tracks passing through a detection D_k . Then $W_{klt}(i, j)$ is the intersection-over-union of the sets S_i and S_j . Essentially $W_{klt}(i, j)$ measures ‘how strongly’ D_i and D_j are connected by point tracks. Figure 9 (right) shows the set of KLT tracks associated to the two UB detections. This affinity measure is more robust than the location one, as it takes into account the motion inside the detection window. This is especially useful when two persons are close in the image, so that their detection windows overlap (see fig. 9, right).

Grouping detections. The three affinity matrices W_{loc} , W_{app} , W_{klt} are combined into a single matrix W_{all} as follows

$$W_{all}(i, j) = W_{loc}(i, j) \cdot W_{app}(i, j) \cdot W_{klt}(i, j) \cdot W_{td}(i, j) \quad (8)$$

$$W_{td}(i, j) = \exp(-(|t_i - t_j| - 1)^2 / \sigma_{td}^2)$$

where t_i , t_j are frame indexes of bounding boxes i , j and $W_{td}(i, j)$ is a damping factor limiting similarity to short time difference. We group detections based on W_{all} using the Clique Partitioning (CP) algorithm of [15], under the constraint that no two detections from the same frame can be grouped. This returns an initial set of tracklets. These tracklets might be interrupted by occlusions, e.g. a person moving behind an occluder would be broken into multiple tracklets (see fig. 10, top).

In a second stage, tracklets are joined together into full long-term tracks based purely on appearance similarity. We define the similarity between two tracklets as the median appearance similarity W_{app} between all pairs of frames in the tracklets and group tracklets using CP. This second stage reconnects tracklets belonging to the same person undergoing occlusion (see fig. 10, bottom).

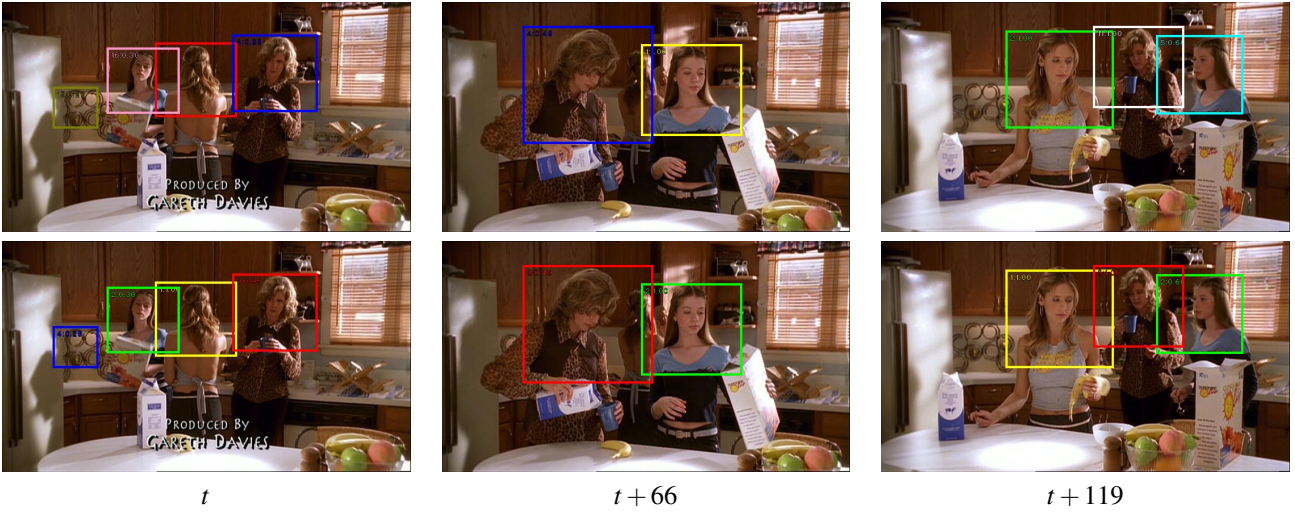


Fig. 10: **Person tracking through full occlusions.** Subsequent frames (t , $t + 66$, $t + 119$) from a video shot where actors swap positions in the scene. Bounding box colors depict different track IDs. Tracks are initially broken into multiple tracklets due to full occlusions (top row). Connecting tracklets based on appearance similarity enables tracking through full occlusion (bottom row).

Post-processing. The process above carefully groups the detections returned by the detector operating in individual frames. However, as the detector is not perfect, it might miss a person in a few frames, even along an otherwise perfect track. In the last stage we fill these ‘holes’ by separate linear interpolation of the window position and size within each tracklet (we assume fixed aspect-ratio of the detection windows). In addition, the position and size values are smoothed over time using a Gaussian filter.

Once the tracking process has finished, false-positive tracks are discarded in a discriminative manner. Inspired by [20], we define a feature vector using the following information: number of detections in the track; ratio of number of detections in the track to the shot length; minimum, maximum, average and standard deviation of the detection scores in the track; minimum, maximum, average and standard deviation of the bounding-box width (note that our UB is squared); absolute and relative ranking position of track in the shot (based on the sum of the detection scores); and, maximum and average overlap (i.e. intersection-over-union) of the track with the other tracks in the same video shot. Then, we train a linear SVM on these feature vectors. For training purposes, we label a track as positive if it goes through a ground-truth bounding-box in at least one video frame (i.e. overlaps at least 0.5 in terms of intersection-over-union). Otherwise, the track is labelled as negative.

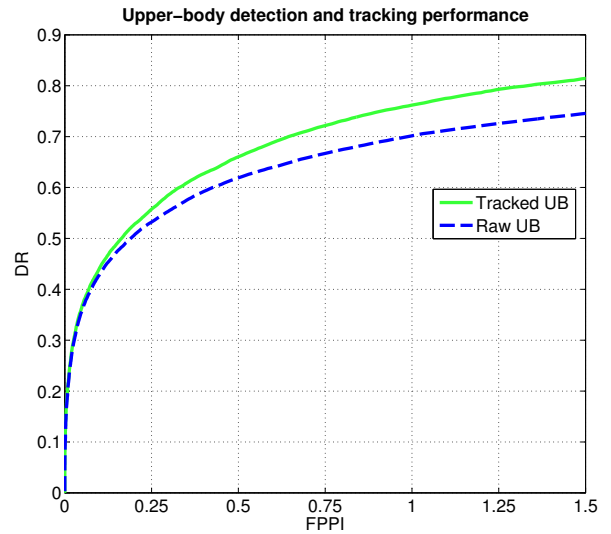


Fig. 11: **Performance of the generic viewpoint upper-body detector and tracker.** Detection rate (DR) versus false positives per image (FPPI) are evaluated on the TVHID dataset. The two curves correspond to detecting upper-bodies independently in each frame (blue) and after tracking (green). The latter uses the entire process described in sec. 4.2, including automatically removing false-positive tracks.



Fig. 12: **Example LAEO labels.** Top-left: label 0 (no LAEO). Top-right: label 1 (clearly visible LAEO). Bottom-left: label 2 (LAEO but with eyes occluded). Bottom-right: label 3 (LAEO but with closed eyes). For our experiments, classes 1, 2 and 3 are considered as positive (green plus symbol), and class 0 as negative (red minus symbol).

4.3 Performance of people detection

We evaluate the performance of the UB detector over test data extracted from the *TV human interactions dataset* (TVHID) of [28]. We evaluate detection rate (DR) versus the average number of false positives per image (FPPI) over all the 27094 frames that compose the dataset. Following the standard PASCAL VOC protocol [12], we count a detection as correct if the intersection-over-union with any ground-truth bounding-box exceeds 0.5.

Fig. 11 reports the performance of our method. The solid green line corresponds to the UBs after tracking, whereas the dashed blue line corresponds to the raw UB detections (i.e. before tracking). The tracker improves DR performance over the whole FPPI range.

Note how our method can handle persons standing close together up to a good degree (see fig. 8). However, for accurately detecting persons in more extreme cases such as when a person is mostly occluded by the other, a separately trained ‘double person detector’ might be necessary, as proposed by [38].

5 Experimental results

5.1 LAEO dataset

We evaluate our LAEO classifiers on the *TV human interactions dataset* (TVHID) of [28]. It contains a total of 300 video clips grouped in five classes: *hand-shake*, *high-five*, *hug*, *kiss* and *negative*. Each video clip might be composed of several shots, and we detect the shot boundaries as maxima in the colour histogram differences between consecutive frames [19].

For our task, we have provided additional annotation for all the videos by assigning one of the following labels to each shot:

- *label 0*: no pairs of people are LAEO
- *label 1*: one or more pairs of people are LAEO in a clearly visible manner
- *label 2*: a pair of people are LAEO, but at least one of them has occluded eyes (e.g. due to viewpoint or hair)
- *label 3*: a pair of people are facing each other, but at least one of them has closed eyes (e.g. during kissing).

There are a total of 443 video shots, where 112 have label 0, 197 label 1, 131 label 2 and 3 label 3. Therefore, the dataset contains 112 negative (label 0) and 331 positive samples (la-

bels 1, 2 and 3). Note that we do not distinguish the three positive labels in the experiments and, for example, we treat looking at each other with closed eyes as a positive. Figure 12 shows an example for each LAEO label. Both the LAEO annotations and the shot boundaries are available at [44].

5.2 Scoring pipeline

We evaluate here the performance of the proposed LAEO classifiers on the following task: *is there any pair of people LAEO at any time in this video shot?*

To assign a LAEO score to a shot we apply the following pipeline:

- (i) assign a LAEO score to each pair of people in every frame using one of the methods in sec. 2. Heads detected by the back-view component are assigned a yaw angle of -135° or $+135^\circ$, depending if they are facing left or right. For the rest of the cases (i.e. frontal/profile head components), the yaw angle returned by the GP regressor is used (i.e. in the range $[-90^\circ, 90^\circ]$);
- (ii) assign a LAEO score to each frame, as the maximum over all pairs of people it contains;
- (iii) slide a window along the temporal axis and average the scores of all frames in the window that are greater than a threshold T ;
- (iv) assign a LAEO score to the shot, as the maximum over all temporal window scores.

Intuitively, these steps will lead to higher scores for pairs of heads that are LAEO over a sustained period of time. This avoids producing false positives for accidental geometric alignments over a few frames (as opposed to simply averaging the thresholded scores over frames). We evaluate performance on the TVHID dataset, using the annotations described above in sec. 5.1. Each method is used to score every shot, and then the average precision (AP) is used to compare the performance of the methods.

5.3 Training-testing setup

The TVHID release of [28] defines two disjoint partitions. We run experiments on two trials, where one partition is used for training and the other for testing, and then report mean AP (mAP) over the two trials.

We set the free parameters of the proposed LAEO scoring methods so as to maximize AP on the training set, using grid search. These parameters are: (i) the aperture ϕ of the cone in the range $[15, 45]$ in steps of 5, for the method of sec. 2.3; (ii) the threshold T on the LAEO scores used by all methods during the temporal window averaging. We tried T the range $[0.2, 0.5]$ in steps of 0.1; and, (iii) the length of the temporal window W in the range $[5, 11]$ in steps of 2.

5.4 LAEO baseline

In addition to the three LAEO methods proposed in sec. 2, we also experiment with a baseline (BL) which, instead of using estimated angles, uses the coarse directional information provided by our head detector (i.e. which model component triggered the detection) to define gaze areas as it in sec. 2.1. Eq. (1) is used to score person pairs. Note that this baseline computes neither yaw nor pitch angles.

5.5 Degrees of automation

In addition to evaluating the proposed LAEO scoring methods, the experiments evaluate the impact of the different stages of our pipeline by replacing them with ground-truth annotations (for the upper-body detector and for the yaw estimator).

Annotated UBs and discretized yaw (GT UB + GT yaw).

In this experiment we use the ground-truth upper-body annotations included in TVHID to estimate the position of the head. We do the following conversion: given an upper-body annotation defined by (x, y, w, h) – top left corner at (x, y) with width w and height h – the estimated head window is computed as $(x + 0.25 \cdot w, y, 0.6 \cdot w, 0.65 \cdot h)$. In addition, the annotated head orientation is used as an approximation of the yaw angle. The following five head orientations are possible in the ground-truth: *profile-left*, *frontal-left*, *frontal-right*, *profile-right* and *backwards*. For our experiments, we map such orientations to the following yaw angles in degrees: -90 , -45 , 45 , 90 and 180 . Since information about pitch angle is not annotated, we set it to 0. Note that TVHID does not contain UB annotations in shots where the UB is not fully visible (i.e. face close-ups). We assign a LAEO score of 0 to such shots.

Annotated UBs with automatic head detection and head pose estimation (GT UB + auto head).

In this experiment we use the annotated upper-bodies included in TVHID to define the tracks of upper-bodies, as in the previous experiment. But all the rest of the processing is automatic (i.e. head detection and head pose estimation). Note that in this experiment we already use the new back-view head detector during the head detection stage.

Fully automatic system. This experiment covers the fully automatic system proposed in this work. We report results for several variants: (i) the system without using the back-view head detector. This corresponds to the results we previously published in [23] (“*Fully auto*”); (ii) the system with the new back-view component of the head detector, used to retrieving more heads only (“*Fully auto + HB*”); (iii) the system using the back-view component to also provide a

rough information about the yaw angle of the head, as discussed in Sec. 4.1 (“Fully auto + HB + BA”).

5.6 Results

Table 2 summarizes the mAP over the two test sets, once the parameters have been optimized over their respective training sets, for each LAEO method separately. Note that in the experiment “GT UB + GT yaw” the baseline method (BL) is not relevant since the head detector is not used (i.e. the components fired by the head detector are needed by BL, but only the ground truth yaw orientation is used in the referred experiment). For placing results in a proper context, the ratio of positive LAEO over the whole dataset is 0.75, and consequently the chance performance of the system is an AP of 0.75.

Figure 13 shows the precision recall curves of the proposed methods for the test set 2 (top row) and test set 1 (bottom row) by using the fully automatic systems without [23] (left) and with (right) back-view head detection (“Fully auto + HB”).

	GA	GC	3D	BL
GT UB + GT yaw	0.869	0.915	0.925	-
GT UB + auto head	0.855	0.893	0.896	0.865
Fully auto [23]	0.822	0.846	0.862	0.816
Fully auto+HB	0.845	0.873	0.876	-
Fully auto+HB+BA	0.841	0.855	0.863	0.842

Table 2: **Summary of LAEO experiments.** Each entry corresponds to the average AP over the test sets. Abbreviations: GA = intersection of gaze areas in 2D (sec. 2.1); GC = geometric constraints in 2D (sec. 2.2); 3D = geometric constraints in 3D (sec. 2.3); BL = baseline (sec. 5.4); HB = head back detector; BA = backview angle.

Discussion. The results reported in table 2 allow different levels of comparison. We can compare different LAEO scoring methods, keeping the degree of automation and the use of the head-back detector fixed (i.e. compare different values within a row). Comparing BL to GA suggest that using the sign of the estimated yaw angles is roughly equivalent to using the coarse head direction represented by which component of the head model triggered the detection (sec. 5.4). The higher performance of GC over GA demonstrates the importance of the information provided by both continuously estimated angles and of the 2D spatial relations between heads. Finally, the most sophisticated LAEO method (3D) consistently delivers the best results in all experiments, although in some cases by a modest amount. The use of a full 3D reasoning (i.e. including the 3D head pose vectors and the

relative position of the people in a 3D coordinate system) is appealing, but we must note that the mAP improvement of the 3D method over GC is small, as shown in the top row of table 2, where the available ground-truth information about people location and their head pose is used.

We can compare performance along another axis by keeping the LAEO scoring method fixed and varying the degree of automation. The better performance of “GT UB + GT yaw” over “GT UB + auto head” highlights the importance of having good head positions and yaw estimates, especially for the GC and 3D LAEO scoring methods. Comparing “GT UB + auto head” to “Fully auto + HB”, we can see that the mAP decreases significantly due to the imperfection of automatic upper-body detection, which include false positives and misses some true positive persons. This fact highlights the importance of using a good person detector and tracker, as the subsequent stages of the pipeline depend on them.

Finally, comparing “Fully auto+HB” to “Fully auto” [23], we can see that mAP is improved for all the LAEO methods. This indicates that detecting back-view heads improves the performance of the system, as it enables to cover more LAEO cases. However, additionally using the angles associated with the back-view components of the head detector does not further improve the LAEO score, since these angles are very imprecise (“Fully auto+HB+BA”).

In summary, the best mAP that we can achieve with a fully automatic method is **0.876**, which is considerably better than both the baseline and chance levels. Our method is able to localise the LAEO pair both spatially and temporally. Figure 14 shows the middle frame of the highest scored temporal window for each of the top 12 ranked shots, according to 3D LAEO scoring method in experiment “Fully automatic+HB”. Note the variety of scenarios where the method successfully works. Only two arguable false positives are present among those 24 video shots.

6 Conclusions

We presented a technique for automatically determining whether people are looking at each other in TV video, including three methods to classify pairs of tracked people. Our best method uses the scale of the detected heads to estimate the depth positioning of the actors, and combines it with the full head pose estimate to derive their gaze volumes in 3D. While we report quantitative performance at shot level, our method allows the interacting people to be localised both spatially (i.e. the pair of heads with the highest LAEO score) and temporally (i.e. temporal sliding window). In conclusion, the recognition of LAEO pairs introduces a new form of high-level reasoning to the broader area of video understanding.

As future work, we plan to study the LAEO problem from the point of view of learning a classifier for LAEO

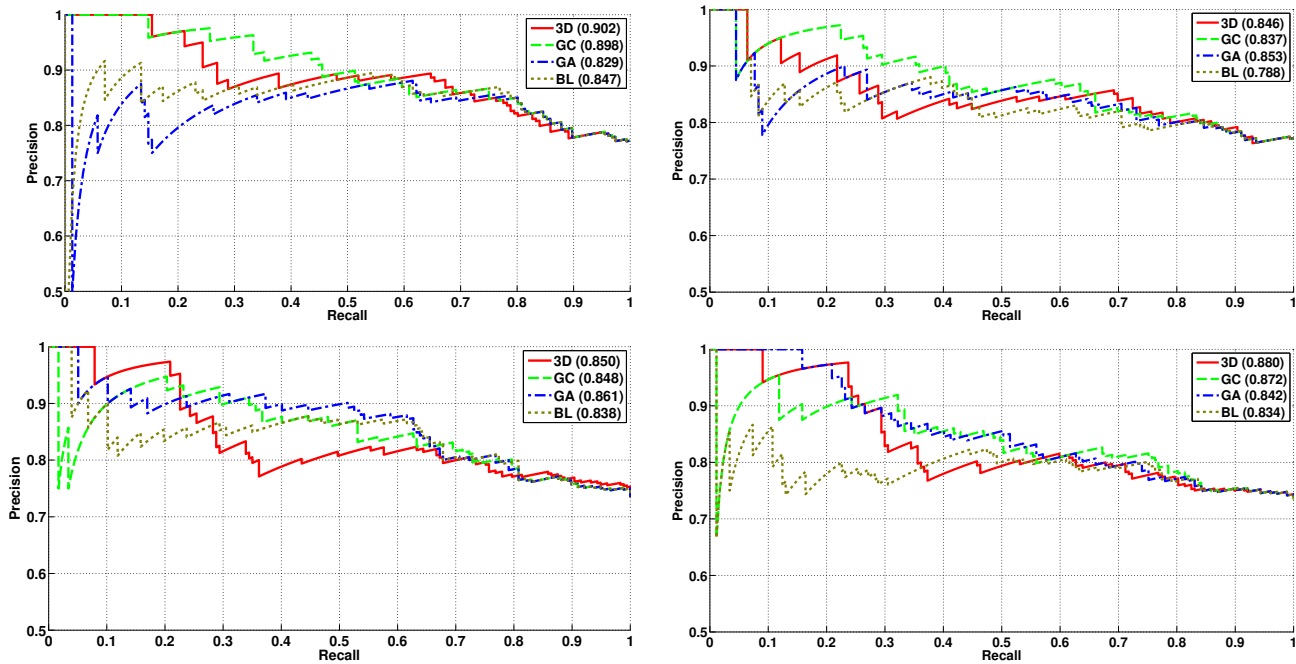


Fig. 13: **Precision recall curves on test sets.** (left) Fully automatic system without back-view head detection [23]. Top: test set 2. Bottom: test set 1. (right) Fully automatic system with back-view head detection (“Fully auto+HB”). Top: test set 2. Bottom: test set 1. In the legend, the AP of each method is shown in parenthesis.

given descriptors over pairs of people as input. For training, this can be cast as a Multiple Instance Learning problem [9].

Acknowledgements We are grateful to the Spanish Minister (projects TIN2012-32952 and BROCA) and for financial support from the Swiss National Science Foundation and ERC grant VisRec no. 228180. We also thank the reviewers for their helpful comments.

Appendix: Released materials

We have released a variety of output from the research that led to this paper: (i) the video shot decomposition [44] of the TVHID videos; (ii) the LAEO annotations [44] on TVHID used in our experiments; and, (iii) the head detector [45] trained to deal with different viewpoints.

References

- Andriluka M, Roth S, Schiele B (2009) Pictorial structures revisited: People detection and articulated pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
- Ba S, Odobez JM (2005) Evaluation of multiple cue head pose estimation algorithms in natural environments. In: Proceedings of the IEEE International Conference on Multimedia and Expo
- Ba S, Odobez JM (2009) Recognizing visual focus of attention from head pose in natural meetings. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Trans on 39(1):16–33
- Benfold B, Reid I (2008) Colour invariant head pose classification in low resolution video. In: Proceedings of the British Machine Vision Conference
- Blanz V, Vetter T (2003) Face recognition based on fitting a 3d morphable model. IEEE Transactions on Pattern Analysis and Machine Intelligence 25:1063–1074
- Bourdev L, Maji S, Brox T, Malik J (2010) Detecting people using mutually consistent poselet activations. In: Proceedings of the European Conference on Computer Vision
- Cour T, Sapp B, Jordan C, Taskar B (2009) Learning from ambiguously labeled images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
- Dalal N, Triggs B (2005) Histogram of Oriented Gradients for Human Detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol 2, pp 886–893
- Dietterich TG, Lathrop RH, Lozano-Prez T (1997) Solving the multiple instance problem with axis-parallel rectangles. Artificial Intelligence 89(12):31–71



Fig. 14: **Test shots ranked according to geometric constraints in 3D.** (Top two rows) Top 12 shots from partition 2 of TVHID, training on partition 1. The frames with red border are (arguable) false positives. (Bottom two rows) Top 12 shots from partition 1 of TVHID, training on partition 2. Note the variety of situations where the proposed method works properly: different scales, poor illumination, cluttered backgrounds, diverse clothing and back-views of people.

10. Everingham M, Zisserman A (2005) Identifying individuals in video by combining generative and discriminative head models. In: Proceedings of the International Conference on Computer Vision
11. Everingham M, Sivic J, Zisserman A (2006) “Hello! My name is... Buffy” – automatic naming of characters in TV video. In: Proceedings of the British Machine Vision Conference
12. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (voc) challenge. International Journal of Computer Vision 88(2):303–338
13. Fathi A, Hodgins J, Regh J (2012) Social interactions: A first-person perspective. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
14. Felzenszwalb P, Girshick R, McAllester D, Ramanan D (2010) Object detection with discriminatively trained part based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9):1627–1645
15. Ferrari V, Tuytelaars T, Van Gool L (2001) Real-time affine region tracking and coplanar grouping. In: Pro-

- ceedings of the IEEE Conference on Computer Vision and Pattern Recognition
16. Ferrari V, Marin M, Zisserman A (2008) Progressive search space reduction for human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 17. Ferrari V, Marin-Jimenez M, Zisserman A (2009) Pose search: Retrieving people using their pose. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 18. Jones M, Viola P (2003) Fast multi-view face detection. Tech. Rep. TR2003-96, MERL
 19. Kim WH, Kim JN (2009) An adaptive shot change detection algorithm using an average of absolute difference histogram within extension sliding window. In: IEEE International Symposium on Consumer Electronics
 20. Kläser A, Marszałek M, Schmid C, Zisserman A (2010) Human focused action localization in video. In: ECCV – International Workshop on Sign, Gesture, Activity, pp 219–233
 21. Laptev I, Marszałek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 22. Liu J, Luo J, Shah M (2009) Recognizing realistic actions from videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 23. Marín-Jiménez M, Zisserman A, Ferrari V (2011) “Here’s looking at you kid”. Detecting people looking at each other in videos. In: Proceedings of the British Machine Vision Conference
 24. Marín-Jiménez M, Pérez de la Blanca N, Mendoza M (2012) Human action recognition from simple feature pooling. In: Pattern Analysis and Applications
 25. Murphy-Chutorian E, Trivedi MM (2009) Head pose estimation in Computer Vision: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence 31:607–626
 26. Osadchy M, Cun Y, Miller M (2007) Synergistic face detection and pose estimation with energy-based models. J Mach Learn Res 8:1197–1215
 27. Park S, Aggarwal J (2004) A hierarchical bayesian network for event recognition of human actions and interactions. Association For Computing Machinery Multimedia Systems Journal
 28. Patron-Perez A, Marszałek M, Reid I, Zisserman A (2010) High Five: Recognising human interactions in TV shows. In: Proceedings of the British Machine Vision Conference
 29. Patron-Perez A, Marszałek M, Reid I, Zisserman A (2012) Structured learning of human interactions in tv shows. IEEE Transactions on Pattern Analysis and Machine Intelligence 34(12):2441–2453
 30. Raptis M, Kokkinos I, Soatto S (2012) Discovering discriminative action parts from mid-level video representations. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 31. Rasmussen CE, Williams CKI (2006) Gaussian Processes for Machine Learning. MIT Press
 32. Sadanand S, Corso J (2012) Action bank: A high-level representation of activity in video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 33. Sapp B, Toshev A, Taskar B (2010) Cascaded models for articulated pose estimation. In: Proceedings of the European Conference on Computer Vision
 34. Shi J, Tomasi C (1994) Good features to track. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp 593–600
 35. Sim T, Baker S, Bsat M (2003) The CMU pose, illumination, and expression database. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(1):1615 – 1618
 36. Sivic J, Everingham M, Zisserman A (2005) Person spotting: Video shot retrieval for face sets. In: Proceedings of the ACM International Conference on Image and Video Retrieval, URL <http://www.robots.ox.ac.uk/~vgg>
 37. Sivic J, Everingham M, Zisserman A (2009) “Who are you?” – learning person specific classifiers from video. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition
 38. Tang S, Andriluka M, Schiele B (2012) Detection and tracking of occluded people. In: Proceedings of the British Machine Vision Conference
 39. Tu Z (2005) Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In: Proceedings of the International Conference on Computer Vision
 40. Waltisberg W, Yao A, Gall J, Gool LV (2010) Variations of a Hough-voting action recognition system. In: Proc. of ICPR 2010 Contests
 41. website (2005) INRIA person dataset. <http://pascal.inrialpes.fr/data/human/>
 42. website (2010) Deformable Parts Model code. <http://www.cs.brown.edu/~pff/latent/>
 43. website (2011) GPML Matlab code. <http://www.gaussianprocess.org/gpml/code/matlab/doc/>
 44. website (2011) LAEO annotations. <http://www.robots.ox.ac.uk/~vgg/data/laeo/>
 45. website (2011) LAEO project. <http://www.robots.ox.ac.uk/~vgg/research/laeo/>
 46. Yang Y, Baker S, Kannan A, Ramanan D (2012) Recognizing proxemics in personal photos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern

Recognition

47. Zhu X, Ramanan D (2012) Face detection, pose estimation and landmark localization in the wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition